Understanding the Scalability of Bayesian Network Inference using Clique Tree Growth Curves

Ole J. Mengshoel
CMU
NASA Ames Research Center
Mail Stop 269-3
Moffett Field, CA 94035
Ole.J.Mengshoel@nasa.gov

Abstract

Bayesian networks (BNs) are used to represent and efficiently compute with multi-variate probability distributions in a wide range of disciplines. One of the main approaches to perform computation in BNs is clique tree clustering and propagation. In this approach, BN computation consists of propagation in a clique tree compiled from a Bayesian network. There is a lack of understanding of how clique tree computation time, and BN computation time in more general, depends on variations in BN size and structure. On the one hand, complexity results tell us that many interesting BN queries are NP-hard or worse to answer, and it is not hard to find application BNs where the clique tree approach in practice cannot be used. On the other hand, it is well-known that tree-structured BNs can be used to answer probabilistic queries in polynomial time. In this article, we develop an approach to characterizing clique tree growth as a function of parameters that can be computed in polynomial time from BNs, specifically: (i) the ratio of the number of a BN's non-root nodes to the number of root nodes, or (ii) the expected number of moral edges in their moral graphs. Our approach is based on combining analytical and experimental results. Analytically, we partition the set of cliques in a clique tree into different sets, and introduce a growth curve for each set. For the special case of bipartite BNs, we consequently have two growth curves, a mixed clique growth curve and a root clique growth curve. In experiments, we systematically increase the degree of the root nodes in bipartite Bayesian networks, and find that root clique growth is well-approximated by Gompertz growth curves. It is believed that this research improves the understanding of the scaling behavior of clique tree clustering, provides a foundation for benchmarking and developing improved BN inference and machine learning algorithms, and presents an aid for analytical trade-off studies of clique tree clustering using growth curves.

1 Introduction

Bayesian networks play a central role in a wide range of automated reasoning applications, including in diagnosis, sensor validation, probabilistic risk analysis, information fusion, and error correction [51, 6, 46, 31, 30, 47, 36]. A crucial issue in reasoning using BNs, as well as in other forms of model-based reasoning, is that of scalability. We know that most BN inference problems are computationally hard in the general case [10, 50, 48, 1], thus there may be reason to be concerned about scalability. One can make progress on the scalability question by studying classes of problem instances analytically and experimentally. Problem instances may come from applications or they may be randomly generated. In the area of application BNs, encouraging as well as discouraging scalability stories have been told. For example, a prominent bipartite BN for medical diagnosis is known to be intractable using current technology [51]. Error correction coding, which can be understood as BN inference, is also not tractable but has empirically been found to be solvable with high reliability using inexact BN techniques [18, 30]. On the other hand, it is well-known that BNs that are tree-structured, including the so-called naive Bayes model, are solvable in polynomial time using exact inference algorithms. There are also encouraging empirical results for application BNs that are "close" to being tree-structured or more generally application BNs that are not highly connected [24, 36].

Clique tree clustering, where inference takes the form of propagation in a clique tree compiled from a Bayesian network (BN), is currently among the most prominent Bayesian network inference algorithms [27, 2, 49]. The performance of tree clustering algorithms depends on a BN's treewidth or the optimal maximal clique size of a BN's induced clique tree [16, 11, 15]. The performance of other exact BN inference algorithms also depends on treewidth.

A key research question is, then, how the clique tree size of a BN (and consequently, inference time) depends on some structural measure of BNs. One way to investigate this is through the use of distributions of problem instances

[52, 5, 11, 41, 21]. Taking this approach, and varying the ratio C/V between the number of leaf nodes C and the number of non-leaf nodes V in BNs, an easy-hard-harder pattern has been observed for clique tree clustering [37].

In this article, we develop a more precise understanding of this easy-hard-harder pattern. This is done by formulating macroscopic and approximate models of clique tree growth by means of restricted growth curves, which we illustrate by using bipartite BNs. Analytically, we consider bipartite BNs created by a random process, the BPART algorithm [37]. The use of a random process represents the fact that exact BN details might not be known (we might in the conceptual design phase) or the fact that there is in fact a setting with a certain amount of randomness to it. For the sake of this work, we then assume that a clique tree propagation algorithm, operating on a clique tree compiled from a BN, is executed in order to answer probabilistic queries of interest. We introduce a random variable for total clique tree size. This random variable is, for the case of bipartite BNs, the sum of two random variables, one for the size of root cliques and one for the size of mixed cliques. Corresponding to the random variable for total clique tree size, we introduce a continuous growth curve for total clique tree size which is the sum of growth curves for the size of root cliques and mixed cliques. A key finding of ours is that Gompertz growth curves are justified on theoretical grounds and also fit very well to experimental data generated using the BPART algorithm [37]. Of particular interest is the growth curve for root clique size, where Gompertz curves of the form $g(\infty)e^{-\zeta e^{-\gamma x}}$ (where $g(\infty)$, ζ, and γ are parameters) turns out to be useful. Our analysis using Gompertz growth curves is novel; they are common in biological and medical research [4, 29] but have not previously been used to characterize clique tree growth. We provide improved analysis compared to previous research, where an easy-hard-harder pattern and approximately exponential growth as a function of C/V-ratio were established [37].

Let W be a random variable representing the number of moral edges in moral graphs induced by random BNs. In addition to $x = C/V$, we consider $x = E(W)$ as an independent variable. In experiments, we compared different growth curves and investigate $x = C/V$ versus $x = E(W)$ as independent variables for Gompertz growth curves. We sampled bipartite BNs using the BPART algorithm. For the number of root nodes, V, we used $V = 20$ and $V = 30$. The number of leaf nodes was also varied, thereby creating BNs of varying hardness. The experimental approach was to randomly generate sample BNs; 100 BNs per C/V-level were generated. A clique tree inference system, employing the minimum fill-in weight heuristic, was used to generate clique trees for the sampled BNs. Linear regression was used to obtain values for the parameters ζ and γ based on a linear form of the Gompertz growth curve; values for $g(\infty)$ were obtained by analysis.

This research is significant for the following reasons. First, analytical growth curves improve the understanding of clique tree clustering's performance. Consider Kepler's three laws of planetary motion, developed using Brahe's observational data of planetary movement. There is a need to develop similar laws for clique tree clustering's performance, and in this article we obtain laws in the form of Gompertz growth curves for certain bipartite BNs [37]. The Gompertz growth curves give significantly better fit to the raw data than alternative curves, provide better insight into the underlying mechanisms of the algorithm, and may be used to approximately predict the performance of clique tree clustering. Our results are thus significant for clique tree clustering, a prominent exact Bayesian network inference algorithm which is studied in detail. Since the performance of other exact BN inference algorithms – including conditioning [44, 11] and elimination algorithms [28, 53, 14] – also depends on optimal maximal clique size, our results may have significance to these algorithms as well. Second, growth curves can be used to summarize performance of different BN inference algorithms or different implementations of the same algorithm on benchmark sets of problem instances, and thereby aid in evaluations. Suppose that the growth curves $g_1(x)$ and $g_2(x)$ were obtained by benchmarking slightly different clique tree algorithms. Compared to looking at and evaluating potentially large amounts of raw data, it may be easier to understand the performance difference between the two algorithms by studying the curves for $g_1(x)$ versus $g_2(x)$ or by comparing their respective parameter values ζ_1 and γ_1 versus ζ_2 and γ_2. Third, growth curves provide estimates of resource consumption in terms of clique tree size. Resource bounds, for example on memory size and inference time, represent requirements from applications and can also be expressed in terms of clique tree size. Hence, this approach enables trade-off studies of resource consumption versus resource bounds, which is important in resource-bounded reasoners [39, 33].

The rest of this article is organized as follows. After introducing notation and background concepts in Section 2, we study the development and growth of BNs, causing corresponding clique tree growth, in Section 3. The issue of independent variables for growth curves, and in particular the C/V-ratio and the expected number of moral edges $E(W)$, is studied in Section 3.1. In Section 3.2, we describe how growth curves can provide a macroscopic model of how clique trees grow as a function of C/V-ratio or expected number of moral edges $E(W)$. In Section 4, we present experiments with varying number of BN root and leaf nodes. We compare different mathematical models of growth, and find that Gompertz growth curves give the best fit to sample data. We conclude and indicate future research directions in Section 5. This article extends and revises an earlier conference paper [34].

2 Background

Graphs, and in particular directed acyclic graphs as introduced in the following definition, play a key role in Bayesian networks.

Definition 1 (Directed acyclic graph (DAG)) *Let* $\boldsymbol{G} = (\boldsymbol{X}, \boldsymbol{E})$ *be a non-empty directed acyclic graph (DAG) with nodes* $\boldsymbol{X} = \{X_1, \ldots, X_n\}$ *for* $n \geq 1$ *and edges* $\boldsymbol{E} = \{E_1, \ldots, E_m\}$ *for* $m \geq 0$*. An ordered tuple* $E_i = (Y, X)$*, where* $0 \leq i \leq m$ *and* $X, Y \in \boldsymbol{X}$*, represents a directed edge from* Y *to* X*.* Π_X *denotes the parents of* X*:* $\Pi_X = \{Y \mid (Y, X) \in \boldsymbol{E}\}$*. Similarly,* Ψ_X *denotes the children of* X*:* $\Psi_X = \{Z \mid (X, Z) \in \boldsymbol{E}\}$*. The out-degree and in-degree of a node* X *is defined as* $o(X) = |\Psi_X|$ *and* $i(X) = |\Pi_X|$ *respectively.*

In the rest of this article we assume that DAGs and BNs are non-empty, even when not explicitly stated as in Definition 1. The following classification of nodes in graphs, including in BNs, turns out to be useful when we discuss the performance of BN inference algorithms.

Definition 2 *Let* $\boldsymbol{G} = (\boldsymbol{X}, \boldsymbol{E})$ *be a non-empty DAG with* $X \in \boldsymbol{X}$*. If* $i(X) = 0$ *then* X *is a root node. If* $i(X) > 0$ *then* X *is a non-root node. If* $i(X) > 0$ *and* $o(X) = 0$ *then* X *is a leaf node. If* $o(X) > 0$ *then* X *is a non-leaf node. If* $o(X) > 0$ *and* $i(X) > 0$ *then* X *is a trunk (non-leaf and non-root) node .*

With the concepts from Definition 2 in hand, we classify the nodes in a DAG as follows.

Definition 3 *Let* $\boldsymbol{G} = (\boldsymbol{X}, \boldsymbol{E})$ *be a DAG. We identify the following subsets of* $\boldsymbol{X}$*:* $\boldsymbol{V} = \{X \in \boldsymbol{X} \mid i\,(X) = 0\}$ *(the root nodes);* $\boldsymbol{C} = \{X \in \boldsymbol{X} \mid i\,(X) > 0$ *and* $o(X) = 0\}$ *(the leaf nodes);* $\boldsymbol{T} = \{X \in \boldsymbol{X} \mid i\,(X) > 0$ *and* $o(X) > 0\}$ *(the trunk nodes);* $\bar{\boldsymbol{V}} = \{X \in \boldsymbol{X} \mid i\,(X) > 0\}$ *(the non-root nodes); and* $\bar{\boldsymbol{C}} = \{X \in \boldsymbol{X} \mid i\,(X) = 0$ *or* $o(X) > 0\}$ *(the non-leaf nodes).*

A Bayesian network (BN) is a DAG with an associated set of conditional probability distributions [45].

Definition 4 (Bayesian network) *A Bayesian network is a tuple* $\beta = (\boldsymbol{X}, \boldsymbol{E}, \boldsymbol{P})$*, where* $(\boldsymbol{X}, \boldsymbol{E})$ *is a DAG with conditional probability distributions* $\boldsymbol{P} = \{\Pr(X_1 \mid \Pi_{X_1}), \ldots, \Pr(X_n \mid \Pi_{X_n})\}$*. Here,* $\Pr(X_i \mid \Pi_{X_i})$ *is the conditional probability distribution for* $X_i \in \boldsymbol{X}$*. Further, let* $n = |\boldsymbol{X}|$ *and let* π_{X_i} *represent the instantiation of the parents* Π_{X_i} *of* X_i*. The independence assumptions encoded in* $(\boldsymbol{X}, \boldsymbol{E})$ *imply the joint probability distribution*

$$\Pr(\boldsymbol{x}) = \Pr(x_1, \ldots, x_n) = \Pr(X_1 = x_1, \ldots, X_n = x_n) = \prod_{i=1}^{n} \Pr(x_i \mid \pi_{X_i}). \tag{1}$$

In this article we will restrict ourselves to discrete random variables, and "BN node" will thus mean "discrete BN node". Let a BN node $X \in \boldsymbol{X}$ have states $\{x_1, \ldots, x_m\}$. We then use the notation $\Omega_X = \Omega(X) = \{x_1, \ldots, x_m\}$ to represent the state space of X. In our discrete setting, a conditional probability distribution $\Pr(X_i \mid \Pi_{X_i})$ is also denoted a conditional probability table (CPT).

A BN is provided input or *evidence* by clamping zero or more of its nodes to their observed states. An instantiation of all non-clamped nodes is an explanation, formally defined as follows.

Definition 5 (Explanation) *Consider a BN* $\beta = (\boldsymbol{X}, \boldsymbol{E}, \boldsymbol{P})$ *with* $\boldsymbol{X} = \{X_1, \ldots, X_n\}$ *and evidence* $\boldsymbol{e} = \{X_1 = x_1, \ldots, X_m = x_m\}$ *where* $0 \leq m < n$*. An explanation* $\boldsymbol{x}$ *is defined as* $\boldsymbol{x} = \{x_{m+1}, \ldots, x_n\} = \{X_{m+1} = x_{m+1}, \ldots, X_n = x_n\}$*.*

When discussing an explanation $\boldsymbol{x}$, the BN β is typically left implicit. Given evidence, answers to different probabilistic queries can be computed by means of a BN. One is often interested in computing answers to queries of the form $\Pr(\boldsymbol{x} \mid \boldsymbol{e})$, and in particular in finding a most probable explanation (MPE). An MPE is an explanation $\boldsymbol{x}^*$ such that $\Pr(\boldsymbol{x}^* \mid \boldsymbol{e}) \geq \Pr(\boldsymbol{x} \mid \boldsymbol{e})$ for any other explanation $\boldsymbol{x}$. In addition to MPE, the computation of posterior marginals (or beliefs) and maximum aposteriori probability (MAP) is of great interest. We distinguish between complete and incomplete algorithms for Bayesian network computation. Complete algorithms include clique tree propagation [27, 2, 23, 49], conditioning [44, 11], variable elimination [28, 53, 14], and arithmetic circuit evaluation [12, 8, 7]. Incomplete algorithms, and in particular stochastic local search algorithms, have been used for MPE [25, 32, 19, 35] as well as MAP [41, 42] computation. Another important distinction is that between algorithms that rely on an off-line compilation step — for example join tree propagation and arithmetic circuit evaluation — and those that do not — for example variable elimination and stochastic local search. Compilation has several benefits when it comes to integration into resource-bounded systems including hard real-time systems [39, 33].

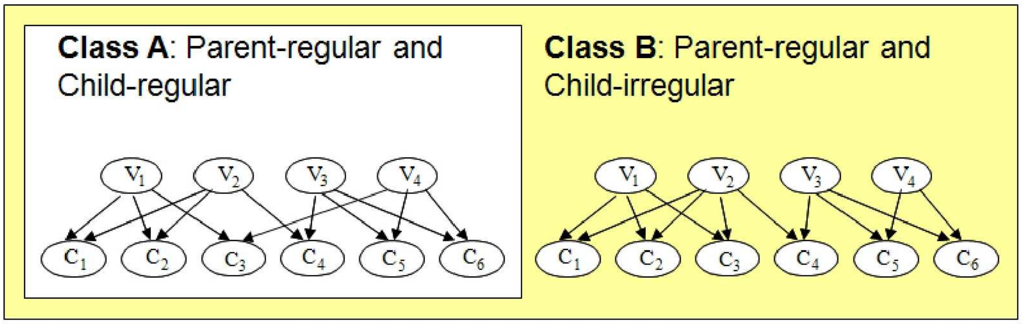
Class A: Parent-regular and Child-regular
V1
V2
V3
V4
C1
C2
C3
C4
C5
C6
Class B: Parent-regular and Child-irregular
V1
V2
V3
V4
C1
C2
C3
C4
C5
C6

Figure 1: Two classes, Class A and Class B, of bipartite graphs and Bayesian networks (BNs). In both Class A and Class B BNs, all leaf nodes have the same number of parents P (here, $P = 2$). In Class A BNs, all root nodes have the same number of children. In Class B BNs, the number of children may vary between the root nodes, as can be seen in this figure.

Our main emphasis in this article is on exact algorithms and in particular compilation using the HUGIN clique tree clustering approach [27, 23]. The HUGIN approach is interesting in its own right, and in addition there is a well-established relationship to arithmetic circuits [43]. A clique tree β''', which is used for on-line computation, is constructed from a BN $\beta = (\boldsymbol{X}, \boldsymbol{E}, \boldsymbol{P})$ in the following way by the HUGIN algorithm [27, 2]. A *moral graph* β' is first constructed by making an undirected copy of β and then augmenting it with moral edges as follows. For each node $X \in \boldsymbol{X}$, HUGIN adds to β' a *moral edge* between each pair of nodes in Π_X if no such edge already exists in β'. Second, HUGIN creates a triangulated graph β'' by heuristically adding fill-in edges to β' such that no chordless cycle of length greater than three exists. Third, a clique tree β''' is created from the triangulated graph β''. A clique tree is created such that for any two nodes F and H in the clique tree, all nodes between them contain $F \cap H$. Using β''', HUGIN can compute marginals [27] or MPEs [13], and the compilation and propagation times are in both cases essentially determined by the size of the clique tree β'''.

The following parameters are useful in characterizing clique trees, and thereby also computation times for algorithms that use clique trees.

Definition 6 (Clique tree parameters) *Let $\boldsymbol{\Gamma} = \{\gamma_1, \ldots, \gamma_\eta\}$ be the set of cliques in a clique tree β'''. The state space size g of a clique $\gamma \in \boldsymbol{\Gamma}$, is defined as*

$$g = |\Omega_\gamma| = \prod_{X \in \gamma} |\Omega_X| \,, \tag{2}$$

where $X \in \boldsymbol{X}$ is a node in $\beta = (\boldsymbol{X}, \boldsymbol{E}, \boldsymbol{P})$. The total clique tree size of β''' (and induced by β) is defined as

$$k = \sum_{\gamma \in \boldsymbol{\Gamma}} |\Omega_\gamma| \,. \tag{3}$$

The performance of many complete BN inference algorithms has been found to depend on treewidth ϖ^* or on optimal maximal clique size ρ^*, where $\varpi^* = \rho^* - 1$ [27, 15]. Treewidth computation is NP-complete [3], and greedy triangulation heuristics that compute upper bounds on treewidth (or optimal maximal clique size) are typically used in practice [26]. A key research question is how treewidth and clique tree size relates to parameters that can be computed for a BN in polynomial time, such as the following parameters:

- $V = |\boldsymbol{V}|$, the number of root nodes in a BN, with $V \geq 1$.
- $T = |\boldsymbol{T}|$, the number of trunk nodes in a BN, with $T \geq 0$.
- $C = |\boldsymbol{C}|$, the number of leaf nodes in a BN, with $C \geq 0$, so the total number of BN nodes is $n = C + V + T$.
- P_{avg}, the average number of parents for all non-root nodes $\bar{\boldsymbol{V}} \subset \boldsymbol{X}$ in a BN, with $1 \leq P_{\text{avg}} \leq N - 1$.
- S_{avg}, the average number of states for all BN nodes $\boldsymbol{X}$, with $S_{\text{avg}} \geq 1$.

Using the parameters above, one can study BN inference and learning algorithms. While our approach is general, we study bipartite BNs in detail in this article. In a bipartite BN $\beta = (\boldsymbol{X}, \boldsymbol{E}, \boldsymbol{P})$, the nodes in $\boldsymbol{X}$ are partitioned into root nodes $\boldsymbol{V}$ and leaf nodes $\boldsymbol{C}$ according to the following definition.

Definition 7 (Bipartite DAG) *Let $\mathbf{G} = (\boldsymbol{X}, \boldsymbol{E})$ be a DAG. If $\boldsymbol{X}$ can be split into partite sets $\mathbf{V} = \{X \in \boldsymbol{X} \mid i(X) = 0\}$ (the root nodes) and $\mathbf{C} = \{X \in \boldsymbol{X} \mid i(X) > 0\}$ (the leaf nodes) such that any $(V, C) \in \mathbf{E}$ is such that $V \in \mathbf{V}$ and $C \in \mathbf{C}$, then $\mathbf{G}$ is a bipartite DAG.*

Important classes of application BNs are bipartite or have bipartite induced subgraphs. Naïve Bayes classifiers are, for example, a special case of bipartite BNs with only one root node. Application areas where bipartite BNs can be found include gas path diagnosis for turbofan jet engines [47], sensor validation and diagnosis of rocket engines [6], diagnosis in computer networks [46], medical diagnosis [51], and error correction [30]. A well-known bipartite BN for medical diagnosis is QMR-DT; in it diseases are root nodes and symptoms are leaf nodes [51]. QMR-DT may be used to compute the most likely instantiation of the disease nodes (i.e., the most probable explanation), given known symptoms [51, 22, 40]. In the area of error correction, a close relationship has been established between error correction decoding in the presence of noise and Bayesian network computation [31, 30]. Specifically, the subgraph induced by nodes corresponding to the hidden information and codeword bits in a decoding BN in fact forms a bipartite BN [31, Figure 7].

In addition, general BNs often have non-trivial bipartite components, and bipartite BNs therefore form a stepping stone for these more general, multi-partite BNs. Bipartite BNs also generalize satisfiability (SAT) instances: root nodes correspond to propositional logic variables and leaf nodes correspond to propositional logic clauses [50, 48, 37]. Special inference algorithms have been designed for bipartite BNs; see for example the study of approximate inference algorithms for bipartite BNs by Ng and Jordan [40].

For the purpose of this article, our main emphasis is on distributions over BNs including randomly generated BNs, as this approach admits a very systematic investigation of BN inference algorithms [52, 21, 37]. Bipartite BNs may be generated randomly using the BPART algorithm [37], which is a generalization of an algorithm that randomly generates hard and easy problem instances for satisfiability [38]. For randomly generated satisfiability (SAT) problem instances, an easy-hard-easy pattern was established as a function of the C/V-ratio for the Davis-Putnam search algorithm [38]. Here, C is the number of propositional clauses and V is the number of propositional variables, and SAT computation is a special case of computing a most probable explanation in BNs [10, 50]. What the easy-hard-easy pattern means is that problem instances go from easy to hard and back to easy again as the C/V-ratio increases. In other words, the hardest problem instances are to be found in the hard region of the easy-hard-easy pattern; this region is also known as the the phase transition region [38, 9, 17].

The BPART algorithm, for which we use the signature BPART(V, C, P, S), operates as follows.[1] First, $V = |\mathbf{V}|$ root nodes and $C = |\mathbf{C}|$ leaf nodes, all with S states, are created. For each leaf node, P parent nodes $\{X_1, \ldots, X_P\}$ are picked uniformly at random without replacement among the V root nodes, creating Class B BNs (see Figure 1). In Class A BNs, which form a strict subset of Class B BNs [37], parents are picked such that all root nodes have exactly k or $k+1$ children for some $k \geq 0$. Conditional probability tables (CPTs) of all nodes are also constructed by BPART; however in this article we focus on the impact of the structural parameters V, C, $P = P_{\text{avg}}$, and $S = S_{\text{avg}}$ on clique tree size. As defaults, parameter values $P = 2$ and $S = 2$ are employed, and we use BPART(V, C) as an abbreviation for BPART(V, C, 2, 2) or when P and S do not matter. Also, we use BPART(V, C, P) as an abbreviation for BPART(V, C, P, 2) or when S does not matter. The total number of edges in a BPART BN is clearly $E = C \times P$.

Here is an example of using clique tree clustering on a small BPART BN.

Example 8 (BPART BN) *Figure 2 shows how a* BPART *BN may be compiled into a clique tree. For each BN leaf node $C \in \{C_1, C_2, C_3, C_4, C_5, C_6\}$, a clique is created. In addition, there are two cliques containing BN root nodes only, namely the cliques $\{V_1, V_2, V_4\}$ and $\{V_2, V_3, V_4\}$.*

Note that tree clustering's moralization step, which creates a moral graph β' from a BPART BN β, ensures that there are edges between all P root nodes that share a leaf node. In order to keep the discussion succinct we often say that BPART creates moral edges without explicitly mentioning tree clustering's moralization step, which actually creates the edges when working on a BPART BN. The processing of the bipartite BN in Figure 2 illustrates the crucial formation of cycles in a BN's moral graph and the resulting generation of fill-in edges. In larger BNs, it is important but also very difficult to understand and predict clique tree clustering's cycle-generation and fill-in processes, which again determine maximal clique size and total clique tree size. A main contribution of this article, further discussed in Section 3 and Section 4, is how we improve the understanding of the growth of total clique tree size as a function of BN growth.

[1] The more extensive signature BPART(Q, F, V, C, S, R, P) was previously used [37]. Here, Q and F are used to control the conditional probability table (CPT) types of BN root and non-root nodes respectively. The parameter R is used to control the regularity in the number of children of root nodes. Since our emphasis in this article is on the impact of the parameters V, C, S, and P, we typically use the default values for Q, F, and R , and also simplify the signature to BPART(V, C, P, S).

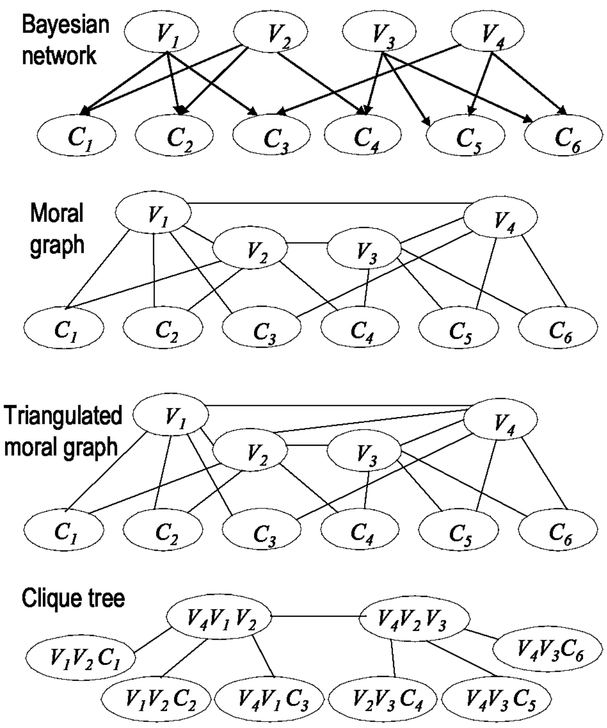

Figure 2: Compilation of BPART BN β (top) to clique tree β''' (bottom). There is a loop (V_1, V_2, V_3, V_4) in the moral graph β', leading to a fill-in edge (V_2, V_4) in triangulated graph β'', which again leads to cliques $\{V_4, V_1, V_2\}$ and $\{V_4, V_2, V_3\}$ in the clique tree β'''.

In the bipartite case, all non-root nodes are leaf nodes (or in other words there are no trunk nodes so $T = 0$) and we have $n = C + V$. We consider only non-empty BNs and so $V \geqslant 1$ and the C/V-ratio is always well-defined. In the important special case of bipartite BNs, the C/V-ratio is the ratio of the number of leaf nodes to the number of root nodes. It has been shown analytically and empirically that the ratio of C to V, the C/V-ratio, is a key parameter for BN inference hardness [37]. Specifically, the C/V-ratio can be used to predict upper and lower bounds on the optimal maximal clique size (or treewidth) of the induced clique tree for BNs randomly generated using the BPART algorithm. Using this approach, upper bounds on optimal maximal clique sizes as well as inference times have been computed [37]. Using regression analysis, the mean number of nodes in the maximal clique was found to be approximately linear in the C/V-ratio. This linear growth translates into an approximately exponential growth in maximal clique size — and consequently in clique tree clustering computation time — as a function of the C/V-ratio. This was found to be true for both Class A and Class B BNs. However, the Class A (or regular) BNs contained maximal cliques that were from 3.0 to 5.4 times larger than maximal cliques in the Class B (or irregular) BNs. By extending previous research on random generation of BN instances, the BPART algorithm provides an approach to easily construct BN instances of varying degrees of difficulty, since the C/V-ratio can be read directly off a BN in linear time, while computing treewidth is NP-hard.

3 Developing Model-Based Reasoners using Bayesian Networks

The development of model-based reasoners, including those that use Bayesian networks, typically involves an iterative or spiral process. One starts with a simple model, which is refined and extended as further information, experimental results, or additional requirements become available. In other words, an iterative development process often manifests itself as model growth, in our case Bayesian network growth. More specifically, if we consider bipartite Bayesian networks used for diagnosis [51, 6, 46, 47], we may distinguish between these two forms of growth:

- Growth in the number of root nodes V, to capture additional faults that may occur in the system being modeled. In a gas path diagnosis BN, these root nodes represent health parameters for a turbofan engine [47], and by increasing the number of health parameters a more comprehensive diagnosis can be computed. In a BN for medical diagnosis, additional root nodes may be introduced because one wants to consider more diseases [51].

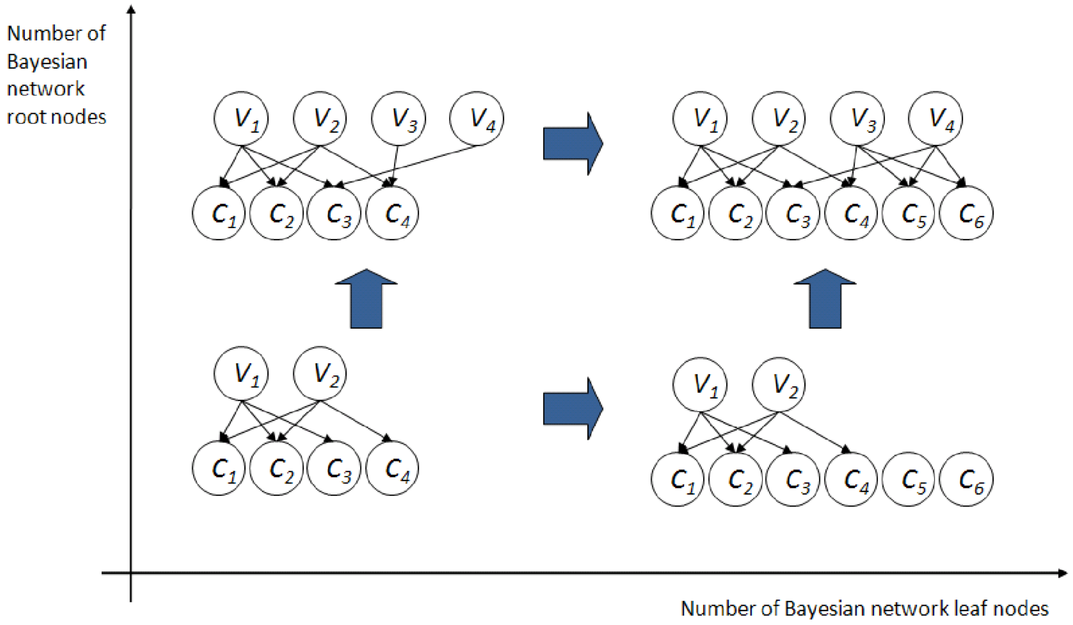
Number of Bayesian network root nodes
V_1
V_2
V_3
V_4
C_1
C_2
C_3
C_4
C_5
C_6
Number of Bayesian network leaf nodes

Figure 3: An example of how a bipartite BN with $V = 4$ root nodes and $C = 6$ leaf nodes (top right) may be developed or grown from a bipartite BN with $V = 2$ root nodes and $C = 4$ leaf nodes (bottom left).

- Growth in the number of leaf nodes C, to represent additional evidence that can be used to distinguish between the underlying faults by computing marginals, MPE, or MAP. In a BN for gas path diagnosis, these leaf nodes can represent additional measurements made on the turbofan engine [47]. In a BN for medical diagnosis, these leaf nodes may represent additional symptoms or tests [51].

A hypothetical BN development process, where small BNs are used for the purpose of illustration, is provided in Figure 3. The figure shows two different BN growth paths leading from a BN with $V = 2$ and $C = 4$ (lower left corner of Figure 3) to a BN with $V = 4$ and $C = 6$ (upper right corner of Figure 3).

Even though we place emphasis on growth or increase here, it is really the concept of change that is important. Our results apply to change in general, both increases and decreases, however the increase or growth perspective is more prevalent. For example, both in knowledge engineering and machine learning one typically develops an application by growing a BN iteratively. In addition, we want to emphasize the connection with biological and medical growth processes [4, 29]. In any case, our work represents a shift away from a particular BN β to families or sequences $(\beta(1), \beta(2), \beta(3), \beta(4), \dots)$ of BNs and the processes by which BNs are developed or grown. The growth processes might be automatic, as in machine learning or data mining, or manual, as in knowledge engineering by direct manipulation of a BN or by using a high-level language from which BNs are auto-generated [36].

An illustration of the connection between BN growth and clique tree growth is provided in Figure 4. It is important to vary a cause (say, the number of leaf nodes in BNs or the density of edges in the moral graphs of BNs) such that a wide range of effects (different clique tree sizes) can be studied. At the highest level, we want to communicate two main ideas in this article. The first idea is the use of a macroscopic growth curve $g_T(x)$ for total clique tree size, where x is an independent parameter. As an illustration, $g_T(x)$ for bipartite BNs is emphasized in Section 3.2, but the approach clearly generalizes beyond bipartite BNs. As a second idea, discussed in Section 3.1, we investigate different independent parameters x in $g_T(x)$. The use of $x = C/V$, where C is number of leaf nodes and V is number of root nodes, is well-known. A novel aspect of this work is the investigation of an alternative to C/V.

The research on the BPART algorithm and its generalization, the MPART algorithm, extends existing research on generating hard instances for the satisfiability problem [38] as well as existing research on randomly generating BNs [52, 5, 25, 11, 41, 20, 21, 37]. Our work on BPART in this article is different from previous research in several ways including the following: The emphasis in this article is on total clique tree size instead of size of largest clique, and in particular we form total clique tree size by partitioning the cliques $\mathbf{\Gamma}$ in the clique tree as discussed in Section 3.2. We closely study the relationship between independent parameters (including C/V) with total clique tree size as the dependent parameter. More specifically, we consider how one can randomly construct Bayesian networks (using BPART) in a controlled way such that the growth of total clique size, as a function of C/V or other independent

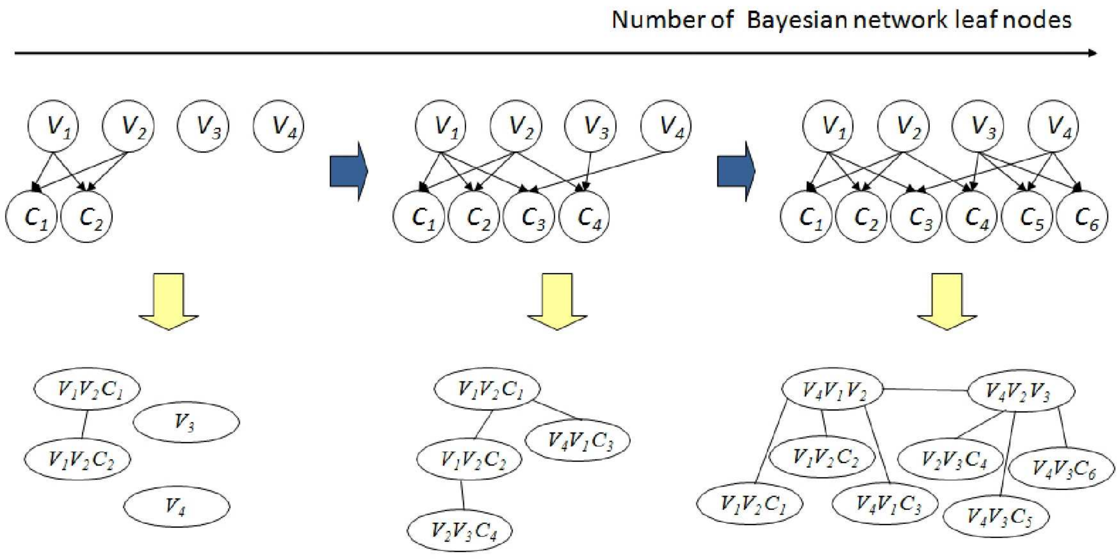

Figure 4: How clique tree size (bottom) varies when the number of BN nodes is varied (top). Horisontally, this figure illustrates how a BN may grow by having leaf nodes added. Vertically, this figure shows how BNs are compiled into clique trees. The growth of a BPART(4, 2) BN (top left) into a BPART(4, 4) BN (top middle) and finally a BPART(4, 6) BN (top right) is depicted in the top row. Clique trees compiled from these BNs are shown in the bottom row.

parameters, can be approximately but reliably predicted.

3.1 Independent Parameters for Bayesian Networks and Moral Graphs

Let W be the random number of moral edges. Then $E(W)$ is the expected number of moral edges. It turns out to be fruitful to use $x = E(W)$ as the independent parameter in $g_T(x)$. In the rest of this section we discuss this issue in more detail.

3.1.1 Balls and Bins

The balls and bins model, where balls are placed uniformly at random into bins, turns out to be useful in our analysis of clique tree clustering's moralization step. Following the balls and bins model, we let m denote the number of balls and n denote the number of bins. Further, we let X and Y be random variables representing the number of empty and occupied bins respectively. The expected number of empty bins X is

$$E(X) = n\,(1 - 1/n)^m\,. \tag{4}$$

The expected number of occupied bins Y is

$$E(Y) = n\,(1 - (1 - 1/n)^m)\,. \tag{5}$$

It is well-established that the expected number of empty bins X can be approximated as

$$E(X) \approx ne^{-m/n}, \tag{6}$$

while the expected number of occupied bins Y is approximated by

$$E(Y) \approx n\left(1 - e^{-m/n}\right). \tag{7}$$

How does the balls and bins model apply to the moral graph created from a BN? The bins are all possible edges in the moral graph, and some nodes induce actual edges (corresponding to balls) in the moral graph. For clarity, we say edge-bin instead of bin and edge-ball instead of ball. The formal definitions are as follows.

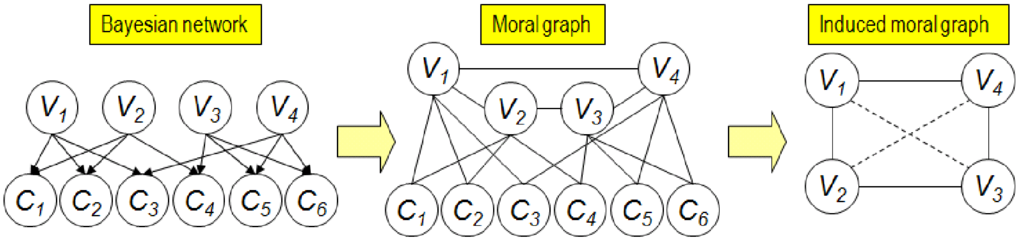

Figure 5: A bipartite Bayesian network (left) is made into a moral graph with moral edges (middle). We focus on the root nodes $\{V_1, V_2, V_3, V_4\}$ and in particular the moral edges in the moral graph's subgraph induced by the root nodes (right). Both the moral edges actually created (edge-bins filled with edge-balls as shown using solid lines) as well as the potential moral edges not created (edge-bins *not* filled with edge-balls as shown using dashed lines) are shown to the right above.

Definition 9 (Edge-bin) *Consider the non-leaf nodes $\bar{\boldsymbol{C}}$ in a BN. An* edge-bin *is a possible edge in the BN's moral graph, between two non-leaf nodes $\{Y_i, Y_j\}$, where $Y_i, Y_j \in \bar{\boldsymbol{C}}$ and $i \neq j$. The set of all* edge-bins *is $\{\{Y_i, Y_j\} \mid Y_i, Y_j \in \bar{\boldsymbol{C}}$ and $i \neq j\}$.*

Definition 10 (Edge-ball) *Consider a non-root node $Z \in \bar{\boldsymbol{V}}$ in a BN, and suppose that $\Pi_Z = \{Y_1, ..., Y_P\} \subseteq \bar{\boldsymbol{C}}$. An* edge-ball *is one edge $\{Y_i, Y_j\}$, where $1 \leq i, j \leq P$ and $i \neq j$, among the set of moral edges $\{\{Y_1, Y_2\}, \{Y_1, Y_3\}, ..., \{Y_{P-1}, Y_P\}\}$ induced by Z.*

In our analysis of clique tree clustering, edge-bins are all possible edges in the moral graph and edge-balls are actual edges induced in a moral graph. We use, as will be seen shortly, a balls and bins approach to obtain the expected number of moral edges in the moral graphs induced by distributions of BNs.

We now consider bipartite BNs where leaf nodes have exactly two parents (Section 3.1.2) or an arbitrary number of parents (Section 3.1.3). For bipartite BNs, the non-leaf nodes are the root nodes and the non-root nodes are the leaf nodes.

3.1.2 Balls and Bins: Two Parents

In the BPART$(V, C, 2)$ model, all edge-bins are uniformly and repeatedly eligible for placing edge-balls into. In other words, we have sampling with replacement. Here is an example of applying our balls and bins model, specifically we consider the number of edge bins for $V = 4$.

Example 11 *Figure 5 shows a BN sampled from the* BPART*(V, C, P, S) distribution with $V = 4$, $C = 6$, and $P = 2$. For this particular BN, 4 of the 6 edge-bins contain edge-balls as can be seen in the subgraph induced by the root nodes $\{V_1, V_2, V_3, V_4\}$ to the right in Figure 5.*

Intuitively, as the C/V-ratio increases, it gets more and more likely that a given moral edge gets picked two or more times, or in other words that an edge-bin contains two or more edge-balls. This intuitive argument is formalized in the following result.

Theorem 12 (Moral edges, exact) *Let the number of moral edges created using* BPART*(V, C, P) be a random variable W. The expected number of moral edges $E(W)$ is, for $P = 2$, given by:*

$$E(W) = \binom{V}{2}\left(1 - \left(1 - 1\Big/\binom{V}{2}\right)^C\right). \tag{8}$$

Proof. We use the balls and bins model. Here, the edge-balls correspond to leaf nodes, of which there are $m = C$. The edge-bins are all possible moral edges, of which there are $n = \binom{V}{2}$ in a bipartite graph with V root nodes. Plugging m and n into (5) gives the desired result (8). ■

It is sometimes convenient to use the following approximation of $E(W)$.

Theorem 13 (Moral edges, approximate) *Let the number of moral edges created using* BPART*(V, C, P) be a random variable W. The expected number of moral edges $E(W)$ is, for $P = 2$, approximated as follows:*

$$E(W) \approx \binom{V}{2}\left(1 - \exp\left(-C\Big/\binom{V}{2}\right)\right). \tag{9}$$

Proof. We use the balls and bins model. Here, the edge-balls correspond to leaf nodes, of which there are $m = C$. The edge-bins are all possible moral edges, of which there are $n = \binom{V}{2}$ in a bipartite graph with V root nodes. Plugging m and n into (7) gives the desired result (9). ■

Given (8) and (9), we can make a few remarks. In contrast to the C/V-ratio or the E/V-ratio, the expectation $E(W)$ takes into account the effect of picking parents among pairs of BN root nodes with replacement. For low values of C/V or E/V one would not expect the effect of replacement to be great, but for large C/V- or E/V-ratios the difference may be substantial as illustrated in the following examples.

Example 14 ($C = 30$ leaf nodes) *Let $V = 30$, $C = 30$, and $P = 2$. The expected number of moral edges is $E(W) = 28.99$ using (8) and $E(W) \approx 29.02$ using (9).*

Example 15 ($C = 300$ leaf nodes) *Let $V = 30$, $C = 300$, and $P = 2$. The expected number of moral edges is $E(W) = 216.91$ using (8) and $E(W) \approx 216.74$ using (9).*

In Example 14, where $E(W) \approx C$, it is unlikely that there are edge-bins with two or more edge-balls. In Example 15, on the other hand, it is very likely that there are edge-bins with two or more edge-balls, and $E(W) < C$. In other words, an additional leaf node has on average a smaller net effect on the number of moral edges in Example 15 than in Example 14, and this is captured in $E(W)$ but not in C/V or E/V. This is important because the difference, as far as cycle (and thus clique) formation in clique tree clustering is concerned, is between (i) no edge-ball and (ii) one or more edge-balls.

3.1.3 Balls and Bins: Arbitrary Number of Parents

We now turn to BPART instances in which P is an arbitrary positive integer. The fundamental complication, as far as the expected number of moral edges $E(W)$ is concerned, is this: For $P > 2$, BPART uses a combination of sampling with replacement and sampling without replacement: Picking the parents of a given leaf node C_i amounts to sampling without replacement, while picking parents for C_i when parents of C_j are already known (for $i > j$) amounts to sampling with replacement. We now introduce, for the purpose of approximation, a variant BPART$^+$ which works exactly as BPART except that the P parent nodes are picked with replacement.

Theorem 16 (Moral edges, exact) *Consider BPART$^+$(C, V, P) and let the number of moral edges created be a random variable Z. The expected number of moral edges is:*

$$E(Z) = \binom{V}{2}\left(1 - \left(1 - 1\Big/\binom{V}{2}\right)^{C\binom{P}{2}}\right). \tag{10}$$

Proof. We use the balls and bins model, and again the number of edge-bins is $n = \binom{V}{2}$ in a bipartite graph with V root nodes. Since BPART$^+$ employs sampling with replacement, the number of edge-balls is $m = C \times \binom{P}{2}$. Plugging m and n into (5) gives the desired result (10). ■

We note that Theorem 16 is a generalization of Theorem 12. Further, we note that $E(Z)$ can be used as an approximation for $E(W)$ for BPART(V, C, P) for $P > 2$, since it is well-known that sampling without replacement can be approximated using sampling with replacement as the number of objects sampled from (here, the V root nodes) grows. Finally, we note that $E(Z)$ in (10) can be approximated in a way similar to the approximation of (8) by (9).

Why are the above balls and bins models of BN moralization interesting? The reason is that we are concerned with the possible *causes*, at a macroscopic level, of clique tree size. The expected number of moral edges is potentially one such cause or independent parameter x. In the context of random BNs, for example as generated by BPART, we indirectly control the placement of moral edges, since we control the Bayesian network's structure through their, for example BPART's, input parameters. The independent parameter x may be defined in terms of a Bayesian network or its moral graph. When it comes to the *effect*, namely tree clustering performance, it is natural to optimize (minimize) the size of the maximal clique. Since this is hard [3], current algorithms including HUGIN use heuristics that upper bound optimal maximal clique size and clique tree size. Such upper bounds on clique tree size are just referred to as clique tree sizes in the following, and we seek in Section 3.2 a closed form expression $y = g(x)$ for the dependent parameter clique tree size as a function of the independent parameter x.

3.2 Dependent Parameters for Clique Trees: Growth Curves

Here, we develop models of restricted clique tree growth that extend exponential growth curves [37] that model unrestricted growth. Even though Bayesian networks and clique trees are discrete structures, we approximate their growth by using continuous growth curves (or growth functions) in order to simplify analysis.

We first discuss bipartite BNs in Section 3.2.1, and then generalize to arbitrary BNs in Section 3.2.2.

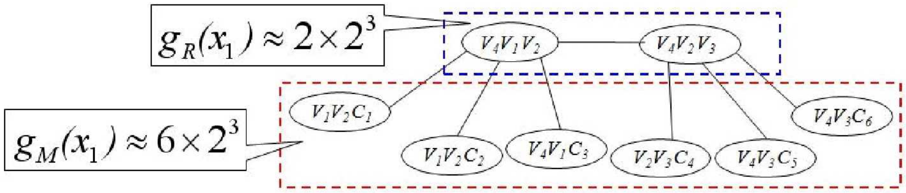

Figure 6: The clique tree for a bipartite Bayesian network, with the two partitions of cliques indicated. Here, $V_4V_1V_2$ and $V_4V_2V_3$ are root cliques while the remaining six are mixed cliques. In addition, g_R is the growth function for root cliques while g_M is the growth function for mixed cliques.

3.2.1 Growth Curves for Bipartite Bayesian Networks

For bipartite BNs, including BPART BNs, there are two types of nodes in the clique tree as reflected in the following definition.

Definition 17 (Root clique, mixed clique) *Consider a clique tree β''' with cliques Γ constructed from a bipartite BN β. A clique $\gamma \in \Gamma$ is denoted a* root clique *if all the BN nodes in γ are root nodes in β. A clique $\gamma \in \Gamma$ is denoted a* mixed clique *if γ contains at least one root node and at least one leaf node.*

It is easy to see that root and mixed cliques are the only clique types induced by bipartite BNs. An illustration of Definition 17 is provided by the clique tree in Figure 6. The BN from which this clique tree is compiled is depicted in Figure 2 and in Figure 4.

We now consider clique trees generated from random BNs. Random variables K_T, K_R, and K_M are used to represent the total clique tree size, the size of all root cliques, and the size of all mixed cliques respectively:

$$K_T = K_R + K_M. \tag{11}$$

Total clique tree size is the sum of the clique sizes of both types, as is appropriate for clique tree clustering algorithms including HUGIN. We use (11) and linearity of expectation to obtain

$$\begin{aligned} E(K_T) &= E(K_R) + E(K_M) \\ \mu_T &= \mu_R + \mu_M. \end{aligned} \tag{12}$$

When varying one or more of BPART's parameters we sometimes make that explicit in (12). For instance, the notation $\mu_R(C)$ or $\mu_M(C)$ means that C is varied while V, P, and S are kept constant. In the experimental part of this article, μ_R will be estimated using its sample mean $\hat{\mu}_R$. Collections of such sample means are then used to construct growth curves.

Given the above vocabulary, we provide a qualitative discussion of the growth of BPART BNs in terms of the C/V-ratio. This discussion is supported by previous (see [37, 34]) and current (see Section 4) experiments and motivates our introduction of growth curves below. In order to keep our discussion relatively simple, we identify three broad stages of clique tree growth, as far as the root cliques are concerned: The initial growth stage, the rapid growth stage, and the saturated growth stage. The *initial growth stage*, where the C/V-ratio is "low", is characterized by "few" leaf nodes relative to the number of root nodes. There is consequently a relatively low contribution by root cliques to the clique tree. This stage is to some extent dominated by mixed cliques — indeed as $C/V \to 0$ there are no root cliques with more than one root node. During the *rapid growth stage*, where the C/V-ratio is "medium", non-trivial root cliques start showing up, due to formation of cycles where fill-in edges are required in order to triangulate the moral graph. An example of the emergence of such a cycle can be seen in Figure 2 and Figure 4. In this stage, and due to the addition of fill-in edges, the root cliques gradually overtake the mixed cliques in terms of their contribution to total clique tree size. The *saturated growth stage*, where the C/V-ratio is "high", is characterized by a "large" number of leaf nodes relative to the number of root nodes. As C/V approaches infinity, one root clique with V BN root nodes (and size S^V in the BPART model) emerges. In this stage, the mixed cliques eventually start to dominate again, since there is one root clique which has reached its maximal size and cannot grow further. However, since the root clique size is exponential in the number of root nodes, it typically takes a long time before the mixed cliques start dominating again, and for non-trivial V this effect can be disregarded.

We emphasize that the discussion above is not entirely formal but is intended as a background for understanding the need for growth curves, which we now introduce.

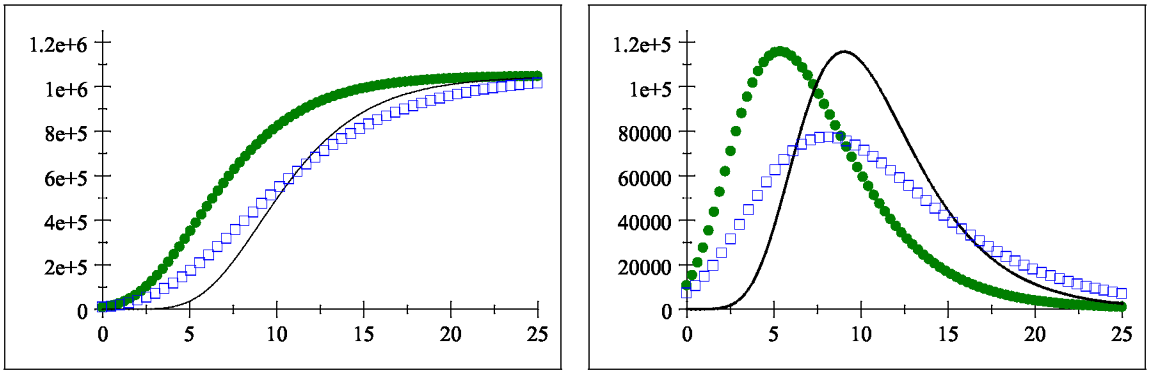

Figure 7: *Left:* Gompertz curves $g_1(x) = 2^{20}e^{-5e^{-0.3x}}$ (green dotted curve), $g_2(x) = 2^{20}e^{-15e^{-0.3x}}$ (black solid curve), and $g_3(x) = 2^{20}e^{-5e^{-0.2x}}$ (blue boxed curve). *Right:* Growth rates $g_1'(x)$, $g_2'(x)$ and $g_3'(x)$ for the Gompertz growth curves.

Definition 18 (Clique tree growth curve) *Let $g_R : \mathbb{R} \rightarrow \mathbb{R}$ and $g_M : \mathbb{R} \rightarrow \mathbb{R}$. Further, let $g_R(x)$ be the growth curve for all root cliques and $g_M(x)$ the growth curve for all mixed cliques. The (total) clique tree growth curve for a bipartite BN is defined as*

$$g_T(x) = g_R(x) + g_M(x).$$

In the following, we will often discuss $g_R(x)$ and $g_M(x)$ independently. In fact, the growth curve for mixed cliques $g_M(x)$ is generally less dramatic than the growth curve for root cliques $g_R(x)$, and therefore we will place more emphasis on $g_R(x)$.

A number of sigmoidal growth curves ("S-curves") have been used to model restricted growth, including the logistic, Gompertz, Complementary Gompertz, and Richards growth curves [4, 29]. For restricted growth curves, $\lim_{x \rightarrow \infty} g(x)$ exists and we define the restricting asymptote as

$$g(\infty) := \lim_{x \rightarrow \infty} g(x). \tag{13}$$

For unrestricted growth curves, including the exponential growth curve, $\lim_{x \rightarrow \infty} g(x)$ does not exist and there is no asymptote $g(\infty)$ as in (13).

It turns out that restricted Gompertz growth curves give very good approximations of root clique growth (see Section 4), and we now study this family of curves in more detail.

Definition 19 (Gompertz growth curve) *Let ζ , $\gamma \in \mathbb{R}$ with $\zeta > 0$ and $\gamma > 0$. A Gompertz growth curve is defined as*

$$g(x) = g(\infty)e^{-\zeta e^{-\gamma x}}. \tag{14}$$

We now discuss some general properties of the form of $g(x)$ in (14). For $x = 0$, clearly $e^{-\gamma x} = 1$, giving $g(0) = (\infty)e^{-\zeta}$ in (14). In other words, the intersection of $g(x)$ with the y-axis is determined by the parameters $g(\infty)$ and ζ in $g(\infty)e^{-\zeta}$. On the other hand, as $x \longrightarrow \infty$ in (14), $e^{-\gamma x}$ tends to 0, meaning that $e^{-\zeta e^{-\gamma x}}$ tends to 1 and thus $\lim_{x \rightarrow \infty} g(x) = g(\infty)$. The greater γ is, the faster $e^{-\gamma x}$ tends to zero, leading to faster convergence to the asymptote $g(\infty)$.

The derivative $g'(x)$ of the Gompertz growth curve

$$g'(x) = \frac{d}{dx}g(x) = g(\infty)\zeta\gamma e^{-\gamma x}e^{-\zeta e^{-\gamma x}}, \tag{15}$$

is an expression of the *growth rate* of $g(x)$; clearly $g'(x) > 0$ given our assumptions in Definition 19.

In Figure 7 we investigate graphically how the parameters $g(\infty)$, ζ, and γ impact the shapes of Gompertz curves. The factor $g(\infty) = 2^{20}$ is obtained, for example, by considering bipartite BNs with $V = 20$ binary ($S = 2$) root nodes. Figure 7 also shows how the growth rate $g'(x)$ changes when the parameters ζ and γ are varied. Let us first vary ζ as shown in Figure 7. By *increasing* ζ from $\zeta = 5$ to $\zeta = 15$ while keeping $\gamma = 0.3$ constant, the x-location of maximal growth rate $g'(x)$ is *increased* as well. However, the value of $g'(x)$ at its maximum does not change. Let us next vary γ as is also illustrated in Figure 7. As γ *decreases* from $\gamma = 0.3$ to $\gamma = 0.2$, while $\zeta = 5$ is kept constant, the x-location of maximal $g'(x)$ *increases*. In addition, the maximal value of $g'(x)$ decreases with γ decreasing, and generally growth gets more gradual as γ decreases.

In the context of BNs, the independent variable x for the growth curve $g(x)$ may be parametrized using $x = C$, $x = C/V$, $x = E/V = CP/V$, or $x = E(W)$, depending on the data available and the purpose of the model. We now introduce, for BPART, a total growth curve model that includes a Gompertz growth curve.

Theorem 20 (BPART Gompertz growth curve) *The total growth curve $g_T(x)$ for* BPART*(V, C, P, S), assuming Gompertz growth for root cliques and where $x = C$ is the independent variable, is*

$$g_T(x) = S^V e^{-\zeta e^{-\gamma x}} + xS^{P+1}. \tag{16}$$

Proof. Since BPART BNs are bipartite, the growth curve has the form $g_T(x) = g_R(x) + g_M(x)$, where $g_R(x) = g_R(\infty)e^{-\zeta e^{-\gamma x}}$ because we have the Gompertz growth curve. For BPART(V, C, P, S) we have $g_R(\infty) = S^V$, and therefore $g_R(x) = S^V e^{-\zeta e^{-\gamma x}}$ for appropriate choices of ζ and γ. Total mixed clique size is $C \times S^{P+1}$ [37], and hence $g_M(x) = xS^{P+1}$. By forming $g_R(x) + g_M(x)$ we obtain the desired result (16). ■

Analytical growth models or growth curves have been used to model growth of organisms and tissue in biology and medicine, growth of technology use or penetration, and growth of organizations or societies including the Web [4, 29]. However, our use of growth curves to model how clique tree size grows with $x = C$, $x = C/V$, or $x = E(W)$ is, to our knowledge, novel. We believe that these macroscopic, coarse-grained models of clique tree growth have benefits that go well beyond strictly experimental results as well as exponential growth curves that model unrestricted growth.

The Gompertz growth curve can be derived by solving the differential equation $dg(x)/dx = ag(x)$, where a is a growth coefficient [4]. Here, a is not constant but exponentially decreasing, formally $da/dx = -ka$ for $k > 0$. These two equations can be solved to obtain (14); see [4]. While a detailed study is beyond the scope of this article, it appears plausible that these differential equations reflect, at a macroscopic level, clique tree clustering's formation of cycles in a moral graph β' along with the generation of fill-in edges. Once one cycle appears in β', there may be many cycles appearing, all needing fill-in edges. Thus, once cycle formation starts in β', a faster than exponential growth in root clique tree size $g_R(x)$ is realistic and indeed supported by previous experimental results [37]. This rapid growth can be captured by Gompertz growth curves.

We emphasize that Gompertz curves do not always provide accurate models of clique tree growth. In particular, the property $g'_R(x) > 0$ does not reflect reality for very small $x = C$. Consider the first few BN leaf nodes added by BPART. When there is no leaf node and $x = 0$, clearly $\mu_R(0) = V$ and $\mu_M(0) = 0$. When there is one leaf node with P parents and $x = 1$, $\mu_R(1) = V - P$ and $\mu_M(1) = S^{P+1}$. Since $\mu_R(0) > \mu_R(1)$, the contribution of the root cliques to the total clique tree size in fact decreases from $x = 0$ to $x = 1$, and clearly this is not consistent with $g'_R(x) > 0$ as follows for example from (15). The situation is similar for other small values of x, see Figure 4 for $x = 2$. However, this early stage of growth is perhaps the least interesting since the total clique tree size is very small and typically not a concern in applications. Consequently, we consider this issue not to be a important limitation of our approach, and in order to circumvent this issue in experiments we use $C/V \geq 1/2$ in our experiments below.

Finally, we note that the Gompertz growth curve has a linear form, defined as follow [29].

Definition 21 (Gompertz linear form) *The Gompertz linear form is*

$$\ln\left(-\ln\frac{g(x)}{g(\infty)}\right) = \ln(\zeta) - \gamma x \tag{17}$$

Using (17), the Gompertz curve parameters ζ and γ in (14) can be estimated from data using linear regression, as we will see in Section 4. Other growth curves, including logistic and Complementary Gompertz, have forms similar to (14) that are also useful for parameter estimation by means of linear regression [29].

3.2.2 Growth Curves for General Bayesian Networks

We now generalize from bipartite BNs to arbitrary BNs. We consider a clique tree $\mathbf{\Gamma}$ generated from an arbitrary BN by clique tree clustering. One way to partition the cliques in a clique tree $\mathbf{\Gamma} = \{\gamma_1, \ldots, \gamma_\eta\}$ is by means of coloring. Formally, we color the nodes in a graph (for us, a BN or a clique tree) as follows.

Definition 22 (Graph coloring) *Let $\boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{E})$ be a graph, let $\Phi = \{1, \ldots, \phi\}$ be a set of colors, and let $h : \boldsymbol{V} \rightarrow \Phi$ be a map (or coloring) from nodes to colors. Then $(\boldsymbol{G}, \Phi, h)$ forms a graph coloring.*

The coloring defines a partitioning of a graph's nodes into ϕ partitions. Definition 22 applies to both directed graphs (including DAGs) and undirected graphs. For BNs we will abuse notation slightly by saying that (β, Φ, h) is a graph coloring when $\beta = (\boldsymbol{X}, \boldsymbol{E}, \boldsymbol{P})$ is a BN; strictly speaking the coloring is in this case only for the DAG part $(\boldsymbol{X}, \boldsymbol{E})$ of the BN.

The following definition of a coloring h partitions nodes into root nodes and non-root nodes.

Definition 23 *Let $\boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{E})$ be a DAG and let $\Phi = \{1, 2\}$ be a set of colors. The coloring $h : \boldsymbol{V} \to \Phi$ reflects the root versus non-root status for any $V \in \boldsymbol{V}$, and is defined as*

$$h(V) = \begin{cases} 1 \, \textit{if} \, i(V) = 0 \\ 2 \, \textit{if} \, i(V) > 0 \end{cases} .$$

Similar to Definition 23, one can define a coloring that partitions nodes into leaf nodes and non-leaf nodes.

How does graph coloring apply to BNs and their clique trees? A clique in a clique tree of a BN β consists of one or more BN nodes, and these nodes may or may not have different colors as induced by a graph coloring (β, Φ, h). To reflect this, we introduce the concept of a color combination for a coloring, and have the following obvious result.

Proposition 24 *Let $(\boldsymbol{G}, \Phi, h)$ be a graph coloring and let $\phi = |\Phi|$. The number of (non-empty) color combinations is*

$$\kappa(\phi) = 2^{\phi} - 1.$$

Similar to Definition 17, we partition the cliques, now according to color combinations. Formally, this amounts to forming subsets of cliques Γ_i for $1 \leq i \leq \kappa(\phi)$ such that $\Gamma = \Gamma_1 \cup \cdots \cup \Gamma_{\kappa(\phi)}$ and $\Gamma_i \cap \Gamma_j = \varnothing$ for $i \neq j$. Assuming that BNs are randomly distributed, we let K_i be the random size of the cliques having the i-th color combination. By summing, we obtain a random variable K_T representing the total clique tree size:

$$K_T = \sum_{i=1}^{\kappa(\phi)} K_i. \tag{18}$$

This sum is a generalization of (11), which applies to the bipartite case.

For each possible color combination, and reflecting the growth of the individual random variables K_i in (18), we introduce a separate growth curve g_i with parameters $\boldsymbol{\theta}_i$ as follows.

Definition 25 *Let (G, Φ, h) be a graph coloring with $\phi = |\Phi|$ and let $g_i : \mathbb{R} \to \mathbb{R}$ be a map with parameters $\boldsymbol{\theta}_i$. Then*

$$g(x; \boldsymbol{\theta}) = \sum_{i=1}^{\kappa(\phi)} g_i(x; \boldsymbol{\theta}_i),$$

where $g : \mathbb{R} \to \mathbb{R}$ and $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_{\kappa(\phi)})$, is a total growth curve.

In words, Definition 25 adds up the growth curves for each color combination. A color combination corresponds to a type of clique. In the bipartite case one could have one color combination for mixed cliques and another color combination for root cliques; see Definition 23. We place no restrictions on the partitioning, but for our purposes it typically makes sense to (i) let the coloring reflect the structure of a graph and (ii) only introduce as many colors as is needed. In this manner, we decompose the problem of estimating growth in total clique tree size into sub-problems of estimating growth curves for smaller pieces of clique trees.

4 Experiments

In the experiments we address the following questions: How well do Gompertz growth curves fit sample data compared to alternative growth curve models? How well do Gompertz growth curves match sample data in the form of clique trees generated from bipartite Bayesian networks using tree clustering, when the independent parameter as well as the nature of the sample data points are varied? In answering these questions, we extend and complement previous experimental results [37] by: (i) introducing restricted growth curves (including restricted Gompertz growth curves) in addition to sample means and unrestricted exponential growth curves; (ii) using a greater range of values for C/V; (iii) considering both $V = 20$ and $V = 30$; (iv) investigating $E(W)$ in addition to C/V as the independent parameter; and (v) using as the dependent parameter the total clique tree size rather than the size of the optimal maximal clique in a clique tree. Clique trees were generated, for sample BNs generated using BPART as indicated below, using an implementation of the HUGIN clique tree clustering algorithm. Clique trees were optimized heuristically, using the minimum fill-in weight triangulation heuristic, as treewidth computation is NP-complete.

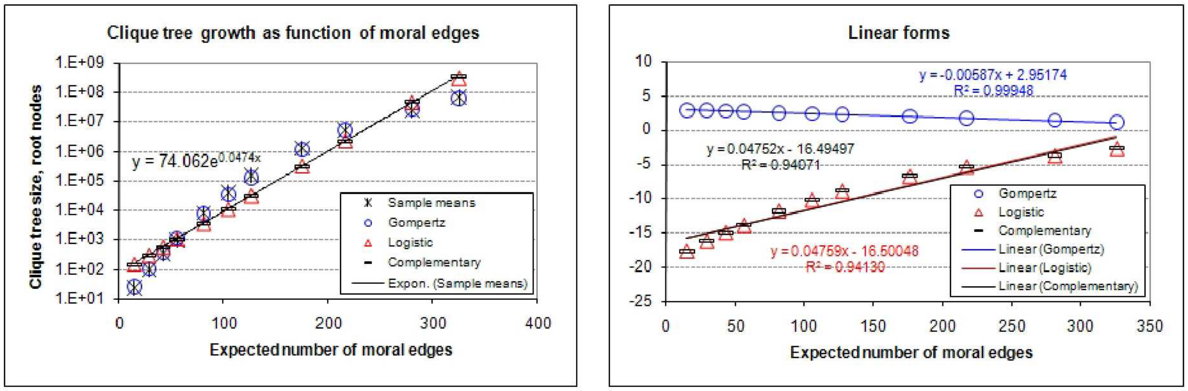

Figure 8: Experimental results for bipartite BNs with $V = 30$ root nodes and varying number of leaf nodes. *Left:* Comparison of Gompertz and other growth curves with the sample means. The superior fit of the Gompertz curve is reflected in its better R^2 value, namely $R^2 = 0.99948$. *Right:* Linear forms showing how the growth curves to the left were obtained.

4.1 Comparison between Growth Models for Multiple BNs

The purpose of the first set of experiments was to compare the Gompertz growth model with a few alternatives: Exponential, logistic, and complementary Gompertz. Here, we report on Bayesian networks generated using the signature BPART(30, C, 2, 2) with varying values for C. For each C/V-level considered, 100 BNs were sampled using BPART.

We now present the results of the HUGIN experiments. In the left panel of Figure 8, sample means $\hat{\mu}_R$ and corresponding points from analytical growth curves as a function of $E(W)$ are presented. The right panel of Figure 8 shows how the growth curves to the left were obtained using linear forms such as (17). The following Gompertz growth curve was obtained

$$g_R(x) = 2^{30} \times \exp(-19.14 \times \exp(-0.005874x)),$$

where $x = E(W)$. The parameters ζ and γ were for the other growth curves computed in a similar manner. Clearly, the Gompertz curve fits the data much better than the alternative growth curves analyzed, with $R^2 = 0.9995$ (for Gompertz) versus $R^2 = 0.9413$ (for logistic) and $R^2 = 0.9407$ (for Complementary Gompertz). The excellent fit can also easily be seen by considering the sample means along with the corresponding data points for the Gompertz curve in the left panel of Figure 8.

4.2 Gompertz Growth Model Details for Multiple BNs

In a second set of experiments, Bayesian networks were generated using BPART(20, C, 2, 2) with varying values for C. For each C/V-level, 100 BNs were sampled using BPART. Using this relatively low value for V allowed us to generate BNs for which the generated clique trees did not exhaust the computer's memory even for very large C, thus supporting a comprehensive analysis using Gompertz growth curves with both $x = C/V$ and $x = E(W)$ as independent variables.

Figure 9 illustrates the results of these experiments. Here, the left column presents Gompertz growth curves, while the right column illustrates how these growth curves were obtained. In the top row of Figure 9, sample means as well as corresponding points from a Gompertz growth curve as a function of C/V-ratio is presented. As a baseline, an exponential interpolation curve for the sample means is also provided. Empirically, the Gompertz growth curve was found to be

$$g(x) = 2^{20} \times \exp(-9.906 \times \exp(-0.1118x)),$$

where $x = C/V$ and with $R^2 = 0.993477$. The values of $\zeta = e^{2.293} = 9.906$ and $\gamma = 0.1118$ were obtained from the Gompertz linear form as illustrated to the top right in Figure 9, based on sample means for the clique tree root cliques and the linear regression result $\ln(\zeta) - \gamma x = -0.1118x + 2.293$.

In the bottom row of Figure 9, we plot the expected number of moral edges $E(W)$ along the x-axis. Note that the right-most sample average in the bottom row of Figure 9, at $x = E(W) \approx 123$, corresponds to the sample average at $C/V = 10$ in the top row of Figure 9. We present sample means along with the corresponding points from a Gompertz

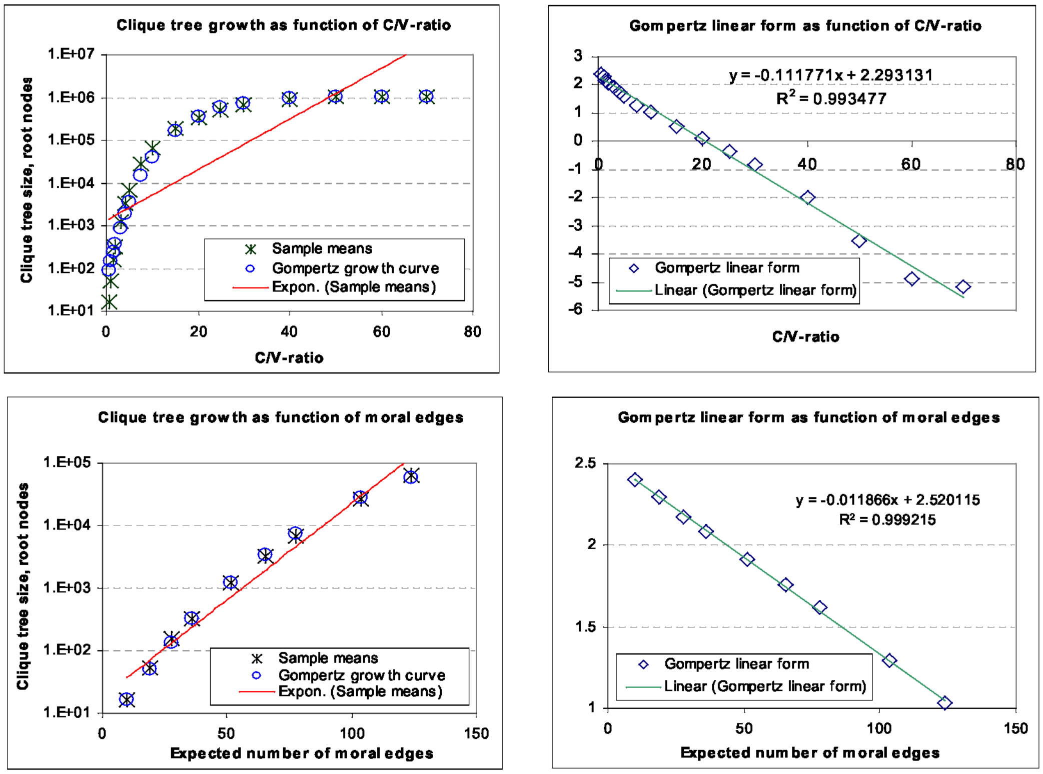

Figure 9: Empirical results for bipartite Bayesian networks generated with $V = 20$ root nodes and a varying number of leaf nodes C. *Top left:* Gompertz growth curve as a function of the C/V-ratio. *Top right:* Gompertz growth curve's linear form as a function of the C/V-ratio; used to create the Gompertz growth curve to the left. *Bottom left:* Gompertz growth curve as a function of $E(W)$. *Bottom right:* Gompertz growth curve's linear form as a function of $E(W)$; used to create the Gompertz growth curve to the left.

growth curve as a function of $E(W)$; an exponential regression curve is presented as a baseline. Here, the Gompertz growth curve was empirically determined to be

$$g_R(x) = 2^{20} \times \exp(-12.43 \times \exp(-0.01187x)),$$

where $x = E(W)$ and with $R^2 = 0.999215$. The parameters ζ and γ were computed in a similar manner to above and as summarized to the bottom right in Figure 9.

We now revisit the three broad growth stages discussed in Section 3 in terms of Figure 9. The sample means show an easy-hard-harder pattern, or monotonically increasing clique tree sizes, along these stages. The *initial growth stage*, where the C/V-ratio is "low" (for $P = 2$, up to approximately $C/V \approx 1$), is characterized by "few" leaf nodes relative to the number of root nodes. In the *rapid growth stage*, the C/V-ratio is "medium" (for $P = 2$, from approximately $C/V \approx 1$ to say $C/V \approx 20$) and non-trivial root cliques appear. As can be seen from the sample means to the left in Figure 9, growth is initially faster than exponential and then slows down. Clearly, the Gompertz growth curves give much better fits than the respective exponential curves for both C/V and $E(W)$. The *saturated growth stage*, where the C/V-ratio is "high", is characterized by slow or no growth due to saturation. At saturation, there is one root clique γ with $|\Omega_\gamma| = 2^{20}$, hence there is no room for further growth. In Figure 9, we may say that saturation starts at $C/V \approx 20$.

Figure 9 clearly shows the improved fit provided by Gompertz curves compared to exponential curves. Further, $x = E(W)$ provides a better fit than $x = C/V$ but for a narrower domain. As a heuristic, one can say that $x = E(W)$ is preferable for local growth models for small values of x, while $x = C/V$ is better for global models and for large x.

4.3 Comparison between Growth Models for Individual BNs

The experimental results so far in this section have been based on sample means of clique tree sizes for BNs. What happens when individual BNs, instead of multiple BNs, are studied using the growth curve approach? To investigate this question, we considered in a third set of experiments BNs generated using the signature BPART(20, C), with C

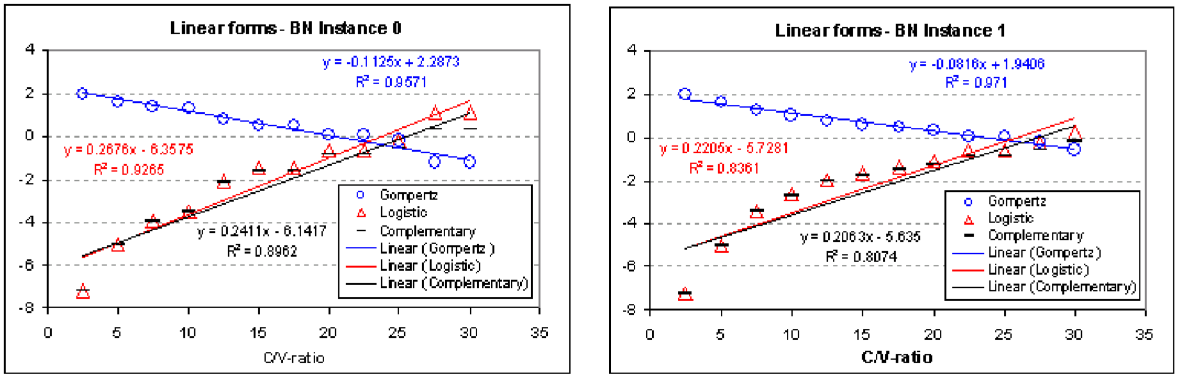

Figure 10: Experimental results for sequences of individual bipartite BNs with $V = 20$ root nodes and a varying number of leaf nodes. Comparison of Gompertz and other growth curves, as a function of C/V, is shown for two sequences of BNs. *Left:* The superior fit of the Gompertz curve for one sequence of BNs is reflected in a higher R^2 value, namely $R^2 = 0.9571$. *Right:* The superior fit of the Gompertz curve for another sequence of BNs is reflected in a higher R^2 value, namely $R^2 = 0.971$.

varying from $C = 100$ to $C = 1200$. The following protocol was followed in order to create a sequence of closely related BNs. Starting with a sampled BPART(20, 1200) BN, 100 leaf nodes were deleted at a time, giving a sequence of BNs consisting of a BPART(20, 1100) BN, a BPART(20, 1000) BN, and so forth, down to a BPART(20, 100) BN. Obviously, in a real development setting the sequence of BNs might be quite different than what we used here, and in particular a machine learning algorithm or a knowledge engineer might start with a small BN and grow it, rather than the other way around. The manner in which the sequence of BNs is created for experimental purposes does not matter as long as they are all BPART BNs, which they clearly are here.

Experimental results for two sequences of BNs, generated according to the above protocol, are presented in Figure 10. This figure clearly shows the better fit provided by Gompertz curves compared to a few alternatives. The better fit is reflected in the higher R^2 values for the Gompertz curves for both sequences. We note that the R^2 values found here, for the Gompertz curves, are smaller than the R^2 values for the Gompertz curves found in Section 4.1 and Section 4.2. A key point in this regard is that each data point here represents the clique tree size of a single BN, while each data point in Section 4.1 and Section 4.2 represents the sample mean clique tree size for 100 BNs. The poorer fit reported here is therefore not surprising.

5 Conclusion and Future Work

Substantial progress has recently been made, both in the area of Bayesian network (BN) reasoning algorithms and in the area of applications of BNs. Based on experience from applications, it is clear that Bayesian networks are useful and powerful but some care is needed when constructing them. In particular, and due to the inherent computational complexity of most interesting BN queries [10, 50, 48, 1], one may want to carefully consider the issue of scalability when developing BN-based reasoning systems for resource-bounded systems including real-time and embedded systems [39, 33]. In resource-bounded systems, BN compilation approaches such as clique tree propagation [27, 2, 23, 49] as well as arithmetic circuit propagation [12, 8, 7] are of particular interest [36]. In the clique tree approach, which we emphasize in this article, BN computation consists of propagation in a clique tree that is compiled from a Bayesian network. Unfortunately, a precise understanding of how varying structural parameters in BNs causes variation in the sizes of induced optimal maximal cliques or clique tree sizes has been lagging. To attack this problem, we have in this article investigated the clique tree clustering approach by employing restricted and unrestricted growth curves. We have characterized the growth of clique tree size as a function of (i) the expected number of moral edges or (ii) the C/V-ratio, where C is the number of leaf nodes and V is the number of non-leaf nodes. In this article, we varied both (i) and (ii) by increasing the number of leaf nodes in bipartite BNs, and also discussed how the approach applies to arbitrary BNs. Gompertz growth curves have, for the bipartite BNs investigated, been shown to give excellent fit to empirical clique tree data and they appear theoretically plausible as well.

The growth curve approach presented in this article and in an earlier paper [34] is novel and extends previous work. In this article, we consider the expected number of moral edges $E(W)$ as well as the C/V-ratio, and a wide range of C/V-ratio values. We focus on the total clique tree size as opposed to size of the largest clique in the clique tree. We believe that the research reported here helps to fill a gap that appears to exist between theoretical complexity results and empirical results for specific algorithms and application BNs. To fill this gap, we have here presented an approach that combines probabilistic analysis, restricted growth curves, and experimentation. Analytically and experimentally, we have shown that the restricted growth curves induce three stages for growing Bayesian networks: The initial growth stage, the rapid growth stage, and the saturated growth stage. Our growth-curve results provide more detail compared to pure complexity-theoretic results; however they admittedly gloss over details available in the raw experimental data.

Areas for future work include the following. First, this type of approach may be utilized in trade-off studies for the design of vehicle health management systems including diagnostic reasoners [33] as well as in the analysis of knowledge-based model construction algorithms. In both cases there is uncertainty regarding the structure of the BNs processed. In knowledge-based model construction, BNs are constructed dynamically, while during the early design of health management systems there may be little information available. Second, these analytical growth curves can also used to perform forecasts and derive requirements for BNs that have clique trees larger than what current software or hardware are capable of supporting. Third, it would be interesting to develop more fine-grained analytical models, perhaps by combining analytical growth models with more extensive data mining.

Acknowledgments

This material is based upon work supported by NASA under awards NCC2-1426, NNA07BB97C, and NNA08205346R.

References

[1] A. M. Abdelbar and S. M. Hedetnieme. Approximating MAPs for belief networks is NP-hard and other theorems. *Artificial Intelligence*, 102:21–38, 1998.

[2] S. K. Andersen, K. G. Olesen, F. V. Jensen, and F. Jensen. HUGIN—a shell for building Bayesian belief universes for expert systems. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, volume 2, pages 1080–1085, Detroit, MI, August 1989.

[3] S. Arnborg, D. G. Corneil, and A. Proskurowski. Complexity of finding embeddings in a k-tree. *SIAM Journal of Algebraic and Discrete Meththods*, 8:277–284, 1987.

[4] R. B. Banks. *Growth and Diffusion Phenomena*. Springer, New York, 1994.

[5] A. Becker and D. Geiger. Approximation algorithms for the loop cutset problem. In *Proceedings of the Tenth Annual Conference on Uncertainty in Artificial Intelligence (UAI-94)*, pages 60–68, San Francisco, CA, 1994.

[6] T. W. Bickmore. A probabilistic approach to sensor data validation. In *AIAA, SAE, ASME, and ASEE 28th Joint Propulsion Conference and Exhibit*, Nashville, TN, 1992.

[7] M. Chavira. *Beyond Treewidth in Probabilistic Inference*. PhD thesis, University of California, Los Angeles, 2007.

[8] M. Chavira and A. Darwiche. Compiling Bayesian networks using variable elimination. In *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence (IJCAI-07)*, pages 2443–2449, Hyderabad, India, 2007.

[9] D. Clark, J. Frank, I. Gent, E. MacIntyre, N. Tomov, and T. Walsh. Local search and the number of solutions. In *Proceedings of the Second International Conference on Principles and Practices of Constraint Programming*, volume 1118 of *LNCS*, pages 119–133, 1996.

[10] F. G. Cooper. The computational complexity of probabilistic inference using Bayesian belief networks. *Artificial Intelligence*, 42:393–405, 1990.

[11] A. Darwiche. Recursive conditioning. *Artificial Intelligence*, 126(1-2):5–41, 2001.

[12] A. Darwiche. A differential approach to inference in Bayesian networks. *Journal of the ACM*, 50(3):280–305, 2003.

[13] A. P. Dawid. Applications of a general propagation algorithm for probabilistic expert systems. *Statistics and Computing*, 2:25–36, 1992.

[14] R. Dechter. Bucket elimination: A unifying framework for reasoning. *Artificial Intelligence*, 113(1-2):41–85, 1999.

[15] R. Dechter and Y. El Fattah. Topological parameters for time-space tradeoff. *Artificial Intelligence*, 125(1-2):93–118, 2001.

[16] R. Dechter and J. Pearl. Network-based heuristics for constraint satisfaction problems. *Artificial Intelligence*, 34(1):1–38, 1987.

[17] J. Frank, P. Cheeseman, and J. Stutz. When gravity fails: Local search topology. *Journal of Artificial Intelligence Research*, 7:249–281, 1997.

[18] R. G. Gallager. Low density parity check codes. *IRE Transactions on Information Theory*, 8:21–28, Jan 1962.

[19] F. Hutter, H. H. Hoos, and T. Stützle. Efficient stochastic local search for MPE solving. In *Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence (IJCAI-05)*, pages 169–174, Edinburgh, Scotland, 2005.

[20] J. S. Ide and F. G. Cozman. Generating random Bayesian networks. In *Proceedings on 16th Brazilian Symposium on Artificial Intelligence*, pages 366–375, Porto de Galinhas, Brazil, November 2002.

[21] J. S. Ide, F. G. Cozman, and F. T. Ramos. Generating random Bayesian networks with constraints on induced width. In *Proceedings of the 16th European Conference on Artificial Intelligence*, pages 323–327, 2004.

[22] T. S. Jaakkola and M. I. Jordan. Variational probabilistic inference and the QMR-DT database. *Journal of Artificial Intelligence Research*, 10:291–322, 1999.

[23] F. V. Jensen, S. L. Lauritzen, and K. G. Olesen. Bayesian updating in causal probabilistic networks by local computations. *SIAM Journal on Computing*, 4:269–282, 1990.

[24] P. Jones, C. Hayes, D. Wilkins, R. Bargar, J. Sniezek, P. Asaro, O. J. Mengshoel, D. Kessler, M. Lucenti, I. Choi, N. Tu, and J. Schlabach. CoRAVEN: Modeling and design of a multimedia intelligent infrastructure for collaborative intelligence analysis. In *Proceedings of the International Conference on Systems, Man, and Cybernetics*, pages 914–919, San Diego, CA, October 1998.

[25] K. Kask and R. Dechter. Stochastic local search for Bayesian networks. In *Proceedings Seventh International Workshop on Artificial Intelligence and Statistics*, Fort Lauderdale, FL, Jan 1999. Morgan Kaufmann.

[26] A. M. C. A. Koster, H. L. Bodlaender, and S. P. M. van Hoesel. Treewidth: Computational experiments. In H. Broersma, U. Faigle, J. Hurink, and S. Pickl, editors, *Electronic Notes in Discrete Mathematics*, volume 8. Elsevier Science Publishers, 2001.

[27] S. Lauritzen and D. J. Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems (with discussion). *Journal of the Royal Statistical Society series B*, 50(2):157–224, 1988.

[28] Z. Li and B. D'Ambrosio. Efficient inference in Bayes nets as a combinatorial optimization problem. *International Journal of Approximate Reasoning*, 11(1):55–81, 1994.

[29] J. K. Lindsey. *Statistical Analysis of Stochastic Processes in Time*. Cambridge, Cambridge, 2004.

[30] D. J. C. MacKay. *Information Theory, Inference and Learning Algorithms*. Cambridge University Press, Cambridge, UK, 2002.

[31] R. J. McEliece, D. J. C. Mackay, and Jung-Fu Cheng. Turbo decoding as an instance of Pearl's belief propagation algorithm. *IEEE Journal on Selected Areas in Communications*, 16(2):140–152, 1998.

[32] O. J. Mengshoel. *Efficient Bayesian Network Inference: Genetic Algorithms, Stochastic Local Search, and Abstraction*. PhD thesis, Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL, April 1999.

[33] O. J. Mengshoel. Designing resource-bounded reasoners using Bayesian networks: System health monitoring and diagnosis. In *Proceedings of the 18th International Workshop on Principles of Diagnosis (DX-07)*, pages 330–337, Nashville, TN, 2007.

[34] O. J. Mengshoel. Macroscopic models of clique tree growth for Bayesian networks. In *Proceedings of the Twenty-Second National Conference on Artificial Intelligence (AAAI-07)*, pages 1256–1262, Vancouver, British Columbia, 2007.

[35] O. J. Mengshoel. Understanding the role of noise in stochastic local search: Analysis and experiments. *Artificial Intelligence*, 172(8-9):955–990, 2008.

[36] O. J. Mengshoel, A. Darwiche, K. Cascio, M. Chavira, S. Poll, and S. Uckun. Diagnosing faults in electrical power systems of spacecraft and aircraft. In *Proceedings of the Twentieth Innovative Applications of Artificial Intelligence Conference (IAAI-08)*, pages 1699–1705, Chicago, IL, 2008.

[37] O. J. Mengshoel, D. C. Wilkins, and D. Roth. Controlled generation of hard and easy Bayesian networks: Impact on maximal clique tree in tree clustering. *Artificial Intelligence*, 170(16-17):1137–1174, 2006.

[38] D. Mitchell, B. Selman, and H. J. Levesque. Hard and easy distributions of SAT problems. In *Proceedings of the Tenth National Conference on Artificial Intelligence (AAAI-92)*, pages 459–465, San Jose, CA, 1992.

[39] D. Musliner, J. Hendler, A. K. Agrawala, E. Durfee, J. K. Strosnider, and C. J. Paul. The challenges of real-time AI. *IEEE Computer*, 28:58–66, January 1995.

[40] A. Y. Ng and M. I. Jordan. Approximate inference algorithms for two-layer Bayesian networks. In *Advances in Neural Information Processing Systems 12 (NIPS-99)*. MIT Press, 2000.

[41] J. D. Park and A. Darwiche. Approximating MAP using local search. In *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence (UAI-01)*, pages 403–410, Seattle, WA, 2001.

[42] J. D. Park and A. Darwiche. Complexity results and approximation strategies for MAP explanations. *Journal of Artificial Intelligence Research (JAIR)*, 21:101–133, 2004.

[43] J. D. Park and A. Darwiche. A differential semantics for jointree algorithms. *Artificial Intelligence*, 156(2):197–216, 2004.

[44] J. Pearl. A constraint - propagation approach to probabilistic reasoning. In L. N. Kanal and J. F. Lemmer, editors, *Uncertainty in Artificial Intelligence*, pages 357–369. Elsevier, Amsterdam, Netherlands, 1986.

[45] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, San Mateo, CA, 1988.

[46] I. Rish, M. Brodie, and S. Ma. Accuracy vs. efficiency trade-offs in probabilistic diagnosis. In *Eighteenth national conference on Artificial intelligence (AAAI-02)*, pages 560–566, Edmonton, Canada, 2002.

[47] C. Romessis and K. Mathioudakis. Bayesian network approach for gas path fault diagnosis. *Journal of engineering for gas turbines and power*, 128(1):64–72, 2006.

[48] D. Roth. On the hardness of approximate reasoning. *Artificial Intelligence*, 82:273–302, 1996.

[49] P. P. Shenoy. A valuation-based language for expert systems. *International Journal of Approximate Reasoning*, 5(3):383–411, 1989.

[50] E. Shimony. Finding MAPs for belief networks is NP-hard. *Artificial Intelligence*, 68:399–410, 1994.

[51] M.A. Shwe, B. Middleton, D.E. Heckerman, M. Henrion, E.J. Horvitz, H.P. Lehmann, and G.F. Cooper. Probabilistic diagnosis using a reformulation of the INTERNIST-1/QMR knowledge base: I. The probabilistic model and inference algorithms. *Methods of Information in Medicine*, 30(4):241–255, 1991.

[52] H. J. Suermondt and G. F. Cooper. Probabilistic inference in multiply connected belief networks using loop cutsets. *International Journal of Approximate Reasoning*, 4:283–306, 1990.

[53] N. L. Zhang and D. Poole. Exploiting causal independence in Bayesian network inference. *Journal of Artificial Intelligence Research*, 5:301–328, 1996.